Please amend the present application as follows:

Claims

The following is a copy of Applicant's claims that identifies language being added with underlining ("___") and language being deleted with strikethrough ("—"), as is applicable:

1. (Currently amended) A method performed by a networked computer peripheral for cycling trigger-event operations of ~~a~~ the networked computer peripheral, the method comprising:

periodically monitoring at least one current timestamp service across a network connection to obtain a current timestamp;

recording a first timestamp indicative of time of a current trigger-event operation;

using the first timestamp and current timestamp, calculating elapsed time since a last trigger-event operation; and

re-running said trigger-event ~~operations~~ operation only when the elapsed time exceeds a pre-specified period for cycling operations.

2. (Original) The method as set forth in claim 1, said calculating further comprising: including accounting for any peripheral power-off time period.

3. (Original) The method as set forth in claim 1, said periodically monitoring further comprising: monitoring coordinated universal time from a network time protocol server.

4. (Currently amended) The method as set forth in claim ~~2~~ 1, comprising:

when no power-off condition occurs during operation of the peripheral, calculating

elapsed time, ET, since a previous trigger-event operation in accordance with an equation:

$$ET = CT_R - TOTE,$$

where CT_R is the current real time and TOTE is a recorded timestamp indicative of the trigger-event, and a determination:

$$\text{is } ET > P_{MAX} ?,$$

where ~~"~~P_{MAX}~~"~~ is a maximum period of operation pre-specified before ~~re- is to be run~~ a predetermined action is to be performed.

5. (Currently amended) The method as set forth in claim ~~2~~ 1, comprising:

when a power off condition occurs during operation of the peripheral, calculating elapsed time since a previous trigger-event operation in accordance with an equation:

$$ET = RT - TOS,$$

where TOS is the approximate time of power off condition, and RT is a timestamp indicative of a following power on condition, and a determination:

$$\text{is } ET > P_{MAX} ?$$

where P_{MAX} is a maximum period of operation pre-specified before a predetermined action is to be performed.

6. (Original) A computer peripheral comprising:

a machine having a memory including a predetermined cyclical machine associated recalibration routine; and

on-board said machine, a network interface coupling the machine to a network, an application for obtaining timestamps across said interface, and a routine for calculating elapsed time since running the predetermined cyclical recalibration routine using said timestamps.

7. (Currently amended) The ~~apparatus~~ peripheral as set forth in claim 6 wherein said calculating includes accounting for time lapsed while said machine is in a power off condition.

8. (Currently amended) The ~~apparatus~~ peripheral as set forth in claim 6 wherein the peripheral is ~~in~~ a computer peripheral printer.

9. (Currently amended) A computer memory having code for cycling calibration operations of a networked computer peripheral, the memory comprising:

computer code configured to periodically ~~monitoring~~ monitor at least one current timestamp service across a network connection to obtain a current timestamp;

computer code configured to ~~recording~~ record a first timestamp indicative of time of current calibration operation;

computer code configured to ~~using~~ use the first timestamp and current timestamp~~, computer code calculating~~ to calculate an elapsed time since a last trigger-event cycle; and

computer code configured to ~~re-running~~ re-run said calibration ~~operations~~ on the computer peripheral only when the elapsed time exceeds a pre-specified period for cycling operations.

10. (Currently amended) The memory as set forth in claim 9, said computer code configured to use the first and current timestamps further being configured to ~~calculating further comprising~~:

~~including accounting~~ account for any peripheral power-off time period.

11. (Currently amended) The memory as set forth in claim 9, said computer code configured to periodically ~~monitoring~~ monitor further ~~comprising~~ being configured to:

~~monitoring~~ monitor coordinated universal time from a network time protocol server.

12. (Currently amended) The memory as set forth in claim ~~10~~ 9, comprising:

when no power-off condition occurs during operation of the peripheral, computer code calculating elapsed time, ET, since a previous calibration operation in accordance with an equation:

$$ET = CT_R - TOTE,$$

where CT_R is the current real time and TOTE is a recorded timestamp indicative of the trigger-event, and a determination:

is $ET > P_{MAX}$?,

where "P_{MAX}" is a maximum period of operation pre-specified before recalibration is to be run.

13. (Currently amended) The memory as set forth in claim ~~10~~ 9, comprising:

when a power off condition occurs during operation of the peripheral, computer code calculating elapsed time since a previous calibration operation in accordance with the equations:

$$ET = RT - TOS,$$

where TOS is the approximate time of power off condition, RT is a timestamp indicative of a following power on condition, and

is $ET > P_{MAX}$?

where P_{MAX} is a maximum period of operation pre-specified before recalibration is to be run.

14. (Original) The memory as set forth in claim 9, the code for cycling calibration operations of a networked computer peripheral comprising: a Java application on a Java Virtual Machine.

15 (New) The method as set forth in claim 1 wherein the trigger-event operation is recalibration of the computer peripheral.

16. (New) The method as set forth in claim 15 wherein recalibration comprises performing a warm-up process on the computer peripheral.

17. (New) The method as set forth in claim 15 wherein recalibration comprises wiping and test-firing ink pens of the computer peripheral.

18. (New) The method as set forth in claim 1 wherein the computer peripheral is a printer.

19. (New) The peripheral as set forth in claim 6 wherein recalibration comprises performing a warm-up process on the computer peripheral.

20. (New) The peripheral as set forth in claim 6 wherein recalibration comprises wiping and test-firing ink pens of the computer peripheral.

21. (New) The memory as set forth in claim 9 wherein the recalibration operation comprises a warm-up process performed on the computer peripheral.

22. (New) The memory as set forth in claim 9 wherein the recalibration operation comprises wiping and test-firing ink pens of the computer peripheral.

23. (New) A method for controlling recalibration of a printer, the method comprising:

obtaining and storing a first timestamp upon completion of a predetermined action performed on the printer using a calibration time routine that executes on the printer;

periodically obtaining and storing current timestamps using the calibration time routine; and

controlling recalibration of the printer based upon the amount of time that has elapsed between the first timestamp and the current time stamps.

24. (New) The method of claim 23, wherein the predetermined action comprises initial booting of the printer.

25. (New) The method of claim 23, wherein the predetermined action comprises printing of a document.

26. (New) The method of claim 23, wherein the predetermined action comprises performance of a calibration process.

27. (New) The method of claim 23, wherein the predetermined action comprises performance of maintenance on the printer.

28. (New) The method of claim 23, wherein the timestamps are obtained from a timestamp resource via a network communication.

29. (New) The method of claim 23, wherein controlling recalibration comprises controlling warm-up of the printer.

30. (New) The method of claim 23, wherein controlling recalibration comprises controlling wiping and test-firing of ink pens of the printer.